OSF/1 AD Version 2 File System Requirements

Paul Roy
David Noveck
Durriya Netterwala

OSF Research Institute
{roy,dnoveck,durriya}@osf.org

July 9, 1993

1 Introduction

This document describes the requirements and goals of the file system for OSF/1 AD Version 2, commonly referred to as AD Version 2 or simply Version 2. Recall that AD Version 2 is targeted for multicomputers and homogeneous clusters, and does not assume the existence of physically shared memory between nodes.

From the file system perspective, the application profile is expected to be a combination of supercomputer applications demainding very high file I/O bandwidth and support for large files, along with more traditional Unix applications, such as compilations, CAD, and word processing.

The distinguishing characteristics of the Version 2 file system will be its system-wide view of the file system name space, including transparent access to remote files and devices, and support for very high bandwidth file I/O. In addition, full multi-node coherency of cached file system information will be provided.

2 Requirements and Goals

Single system image name space. There must be a single name space for all nodes in the system, including a system-wide root. Management of the name space by system administrators must be straight-forward, including the ability to encapsulate information such as the the physical location of devices, where devices are mounted, etc. in a single file system partition, and the ability to manipulate or query the file system name space with familiar commands.

It is a goal that third parties be able to adapt Version 2 to a multiple root environment (i.e., each node has its own root), but it is not a requirement that Version 2 implement such a configuration.

Remote device access. It must be possible for an application to access devices attached to remote nodes. This includes the ability to have distinct file servers managing the name of a device (the device special file) and the device itself. In addition, a file server must be able to manage a device attached to a remote node, utilizing Mach's network-transparent device interface.

Scalable. The file system must not contain bottlenecks when scaled to thousands of nodes.

High bandwidth file access. There must be a high bandwidth access path to files stored on remote file server nodes. The overhead due to the file system code itself should be sufficiently low such that the overall file I/O performance is comparable to Version 1's fast path, and will improve in accordance with the performance improvements made to NORMA IPC and/or the device subsystem.

This access path need not provide compute-node caching of file system information.

Cached file access. AD Version 2 must support compute-node caching of file system data. The intent is that access to already-cached data will not suffer the delays inherent to contacting a remote server. In addition, it is a goal to support compute-node caching of file atributes and pathnames. All caching should be provided on a multiple-reader/single-writer basis.

The performance of cached file access should be better than AD Version 1, particularly through the use of bulk operations for transferring data between disk and cache.

It is not a requirement that a given file support both high bandwidth and cached access methods, but rather they may be provided on a per-mounted file system basis.

Large files and file systems. AD Version 2 must support files larger than 2Gb, most likely through the use of 64 bit file offsets and file sizes. Further design work is required before determining the exact nature of this suppport.

High performance local file system. AD Version 2's default local file system, the UFS, must support high bandwidth access (no data caching, no copies, write-behind), large contiguous data transfers to and from disk (on the order of 512 Kb or 1 Mb), and disk block preallocation.

Variable blocks sizes. AD Version 2 must support disks and file systems with different sector sizes and file system blocks sizes, respectively.

Support for distributed process execution. AD Version 2's file system must support processes executing on different nodes. Of particular note is the need to support shared file offsets between such processes.

Configurable compute nodes. To satisfy the needs of vendor or site-specific configurations, it must be possible to configure compute nodes with minimal file system functionality. In particular, it must be possible to "configure out" the cached method of file access as well as the client-side remote device code with minimal modifications to other code, although compile-time configuration is acceptable.

Failure containment. The failure of a file server may result in the portion of the name space managed by it, or by a server managing file systems "below" it in the same branch, to become inaccessible. However, it mustn't affect accessibility to other portions of the file system tree. In addition, failed client-side or server-side file system servers must allow other servers to cleanup relevant state through appropriate use of Mach kernel functionality such as dead-name notifications.

Note that cleanup from a failed server might be required at the Mach layer as well (e.g., cleaning up an XMM connection).

Multiprocessor efficient. The AD Version 2 file system design must take into account MP issues, including efficient exectuion. If AD Version 2 is ported to a multiprocessor, then the implementation must be verified in this regard.

Quality implementation. The implementation must meet high standards of code quality, maintainability, and robustness. Although the OSF Research Institute itself is not a *product* organization, it must be possible to productize AD Version 2 in general, and the file system in particular, in a short amount of time (e.g., a few months).

Modular implementation. The implementation must meet high standards of modularity. In particular, file system code should be well-encapsulated and a genreal vnode interposition mechanism should be proveded to support modular development. It is not a goal to develop a full stackable file system framework.

OSF/1 API compliance. The system must adhere to the complete OSF/1 R1.3 API. Extensions to the API will also be provided as necessary for the target environment.

Binary compatibility. It is a goal that AD Version 2 commands and libraries be binary compatible with those from AD Version 1, although it is not an absolute requirement. At the least, AD Version 2 should strive to avoid unnecessary incompatibilities relative to AD Version 1. Similarly, it is a goal for AD Version 2 to be binary compatible with OSF/1 R1.3, but not an absolute requirement.

File system interoperability. AD Version 2 must allow a multicomputer or cluster to interoperate with NFS, either as a client or a server. In addition, it must provide AFS client support. This is the same functionality currently provided by AD Version 1.

B1 Security. Although AD Version 2 will not implement support for B1 security, it mustn't preclude it from being implemented with relative ease.

Flexible and extensible. The AD Version 2 file system must be flexible and extensible to meet a variety of vendor-specific requirements. This includes the ability to easily add:

- other local file systems
- different caching strategies
- striped or parallel file systems
- replicated file systems

It is also a goal that a vendor be able to replace AD Version 2's distributed file system functionality without undue burden.

3 Non-Goals

- Absolute source or binary compatibility with AD Version 1 or OSF/1 R1.3.
- Full stackable file system framework support.
- Explicit support for database or OLTP environments.
- Support for the Memory File System (MFS), Logical Volume Manager (LVM), or System V file system (S5FS).